Simple Deep Random Model Ensemble

Xiao-Lei Zhang, *Member, IEEE,* and Ji Wu, *Member, IEEE*

Abstract—Representation learning and unsupervised learning are two central topics of machine learning and signal processing. Deep learning is one of the most effective unsupervised representation learning approach. The main contributions of this paper to the topics are as follows. (i) We propose to view the representative deep learning approaches as special cases of the knowledge reuse framework of clustering ensemble. (ii) We propose to view sparse coding when used as a feature encoder as the consensus function of clustering ensemble, and view dictionary learning as the training process of the base clusterings of clustering ensemble. (ii) Based on the above two views, we propose a very simple deep learning algorithm, named deep random model ensemble (DRME). It is a stack of random model ensembles. Each random model ensemble is a special k-means ensemble that discards the expectation-maximization optimization of each base k-means but only preserves the default initialization method of the base k-means. (iv) We propose to select the most powerful representation among the layers by applying DRME to clustering where the single-linkage is used as the clustering algorithm. Moreover, the DRME based clustering can also detect the number of the natural clusters accurately. Extensive experimental comparisons with 5 representation learning methods on 19 benchmark data sets demonstrate the effectiveness of DRME.

Index Terms—Clustering, deep learning, dictionary learning, ensemble learning, sparse coding, unsupervised representation learning.

1 Introduction

REPRESENTATION learning is to learn transformations of the data that makes it easier to extract useful information when building classifiers or other predictors [1]. Popular representation learning techniques include ICA in source separation, PCA in dimension reduction, kernel learning in classification, and Bayesian nonparametric models in data modelling. As was argued by Hinton *et al.* [2], these methods are all *shallow* models that learn linear or only one layer of nonlinear transformations, so that (i) their representative powers are limited, (ii) the numbers of their parameters grow rapidly with the size of the data set, or (iii) they have both of the aforementioned weaknesses. Therefore, the *deep* models, which contain multiple layers of nonlinear transformations, are suggested as one of the recent advances. The main advantage of the deep models over shallow ones lies in that "*functions that can be compactly represented by a depth k architecture might require an exponential number of computational elements to be represented by a depth $k-1$ architecture*" [3].

The main difficulty of the deep models is that multiple layers of nonlinear transformations make the models suffer severely from bad local minima. In 2006, a breakthrough of training the deep models was made by Hinton *et al.* [2], followed by revolutionary improvements on image processing and speech recognition [4], [5].

- *All authors are with the Multimedia Signal and Intelligent Information Processing Laboratory, Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing, China, 100084.*
  E-mail: huoshan6@126.com, wuji_ee@tsinghua.edu.cn

This work was supported in part by the National Natural Science Funds of China under Grant 61170197 and in part by the China Postdoctoral Science Foundation funded project under Grant 2012M520278.

Currently, the successful training method of deep belief networks (DBN) in [2] becomes a standard one. It consists of two phases – the unsupervised greedy layer-wise pre-training phase and the supervised fine-tuning phase. The pre-training phase is the key idea of deep learning that helps the deep models get rid of bad local minima. It is also an active area the researchers enjoy in. The pre-training phase aims to train a stack of shallow modules successively, where the input data of each module is the output of its ancestor (i.e. previous) module. The representative shallow modules include the restricted boltzman machine (RBM) [2] and denoising autoencoder (DAE) [6], [7]. See [1], [3] for excellent reviews.

However, deep learning is far from explored and understood yet. In this paper, we pay attention to the following two key respects. First, current discussions on deep learning are still limited to probabilistic graphic models and neural networks, other meaningful interpretations and successful building blocks are seldom seen. Second, existing deep models are still too complicated for a wide range of applications. As we known, a widely used method should be simple and fast, such as the k-means clustering. Also, currently, there is a trend of simplifying the state-of-the-art modeling techniques for efficiency, such as using k-means to learn feature representations [8] and discussing the relationship between the Dirichlet process and the k-means [9], [10]. For the above two respects, we focus on discussing the following two problems:

- How to understand the success of the unsupervised pre-training of deep learning, so as to guide the design of new building blocks?
- Can we get a very simple deep learning method that a freshman can play with?

For problem 1, (i) we view the existing successful building blocks [2], [6], [7] in the perspective of *ensemble learning*, and view the unsupervised layer-wised pre-training phase as a knowledge reuse framework of *clustering ensemble*, so that a vast amount of ensemble learning techniques are available for designing new building blocks. Ensemble learning [11] is an important branch of machine learning that aims to combine a serial base learners for a stronger one. Clustering ensemble is the unsupervised extension of ensemble learning [12]–[15]. The success of ensemble learning are supported by two basic criteria – meaningful base learners and strong diversity among the base learners. See Section 4.1 for a further introduction. (ii) We further pay particular attention to the important experimental phenomena in [16] on sparse coding, where sparse coding is an important shallow representation learning approach. If we view sparse coding (when used as a feature encoder) as the consensus function of clustering ensemble, and if we view dictionary learning as the training process of clustering ensemble, we can explain the success of the very simple sparse coding methods in [16] easily.

For problem 2, we first use the k-means ensemble [13] as the building block of a deep architecture. Because the k-means ensemble is very inefficient, we discard the expectation-maximization (EM) optimization of the k-means but only preserve the default initialization method of the k centers of the k-means – random observation sampling. The proposed DRME contributes to such a great simplification of deep learning that it even does not need an obvious optimization objective and does not need any sophisticated optimization algorithm. Although the proposed algorithm is so simple, it performs surprisingly well in practice, such as our application to clustering.

The key idea why we discard the EM optimization are motivated step by step as follows: (i) After viewing the building blocks of the representative deep learning approaches as special cases of clustering ensemble, we take the two basic criteria of clustering ensemble as our design criterions. One key criterion is how to train a meaningful base clustering. (ii) After viewing the successful approximation of the contrastive-divergence (CD) training [17], [18] to maximum likelihood training for DBN, we find that even reducing the maximum iteration number of the EM training gradually from a large number to zero, the randomly sampled k-centers can still be a meaningful base clustering. (iii) After explaining the confidential experimental phenomena of sparse coding in [16] in the perspective of clustering ensemble and further building a relationship between the work in [16] and the proposed DRME, we find a strong empirical support of the proposed DRME in literature.

The main contributions are summarized as follows.

- We view the representative deep learning approaches [2], [6], [7] as special cases of the knowledge reuse framework of clustering ensemble [12].
- We explain the success of the simple sparse coding approaches when used as feature encoders in [16] in the perspective of clustering ensemble.
- We propose a very simple and fast deep learning algorithm, called DRME.
- We propose a new scheme on how to find the most powerful representation among the layers by applying DRME to clustering. The DRME based clustering, as a by-product, can also detect the number of the natural clusters automatically [12], [13], [19], which is a well-known hard problem of clustering.
- We conduct an extensive experimental comparison with 5 state-of-the-art unsupervised representation learning algorithms on 19 benchmark data sets.

The remainder of this paper is organized as follows. In Section 2, we will present the proposed DRME algorithm "suddenly" so as to give the reader a first image on how simple our algorithm is. In Section 3, we will apply DRME to clustering. In Section 4, we will review three related topics for preparing the discussion on our motivation in Section 5, where the three related topics are clustering ensemble, deep learning, and sparse coding, respectively. In Section 5, we will first present how we view the popular deep learning methods as stacked clustering ensembles, and then explain why we can reduce the clustering ensemble to the random model ensemble. In Section 6, we will conduct an extensive experimental comparison, where the performance is evaluated by the clustering accuracy and running time. At last, in Section 7, we will conclude this paper.

We first introduce some notations here. Bold small letters, e.g., $\mathbf{w}$ and $\boldsymbol{\alpha}$, indicate column vectors. Bold capital letters, e.g., $\mathbf{W}$, $\mathbf{K}$, indicate matrices. Letters in calligraphic bold fonts, e.g., $\mathcal{A}$, $\mathcal{B}$, and $\mathbb{R}$, indicate sets, where $\mathbb{R}^d$ denotes a d-dimensional real space. The operator $\|\cdot\|_m$ denotes the m-norm, where m is a constant.

2 Deep Random Model Ensemble

In this section, we will first review the key idea of deep learning. Then, we will present the deep random model ensemble and analyze its time and space complexities. At last, we will illustrate the effectiveness of the proposed method on a handwritten digit recognition problem.

2.1 Preliminary

For the unsupervised representation learning, we are interested in learning a mapping f_θ from input $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ to a novel representation $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$, i.e. $\mathbf{y}_i = f_\theta(\mathbf{x}_i), \forall i = 1, \ldots, n$, where $\mathbf{x}_i \in \mathbb{R}^d$, $\mathbf{y}_i \in \mathbb{R}^D$, and θ is the parameter of the mapping function.

For a deep architecture with L layers, we aim to learn $\mathbf{Y}$ through L mapping functions $\{f_{\theta_l}\}_{l=1}^L$, i.e. $\mathbf{y}_i = f_{\theta_L}\left(f_{\theta_{L-1}}\left(\ldots f_{\theta_1}(\mathbf{x}_i)\right)\right), \forall i = 1, \ldots, n$.

For the unsupervised layer-wise training of a deep architecture, we train each mapping function independently with the input of the mapping as the output of its

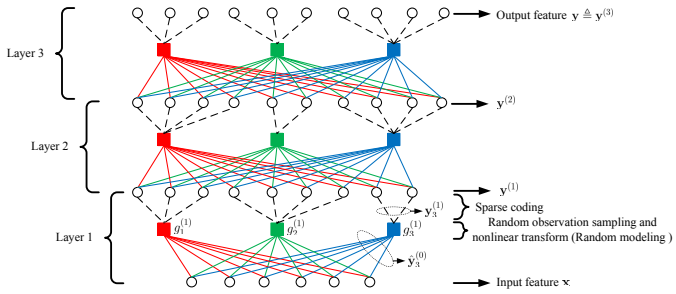

Fig. 1. Diagram of the architecture of the proposed deep random model ensemble.

ancestor mapping function, which can be formulated as a problem of learning the following functions successively:

$$\begin{aligned} \mathbf{y}_i^{(1)} &= f_{\theta_1}(\mathbf{x}_i), \\ \mathbf{y}_i^{(2)} &= f_{\theta_2}\left(\mathbf{y}_i^{(1)}\right), \\ &\vdots \\ \mathbf{y}_i \triangleq \mathbf{y}_i^{(L)} &= f_{\theta_L}\left(\mathbf{y}_i^{(L-1)}\right), \quad \forall i = 1, \ldots, n \end{aligned} \tag{1}$$

where θ_l and $\mathbf{y}^{(l)}$ are the parameter and output representation of the l-th layer respectively with $l = 1, \ldots, L$ and $\mathbf{y}^{(l)} \in \mathbb{R}^{d_l}$.

2.2 Algorithm Description

The key idea of the proposed DRME is to stack multiple random model ensembles, where the random model ensemble is a reduced k-means ensemble [13] that preserves the default initialization method (i.e. random observation sampling) of the k-centers of each base k-means clustering but discards the EM optimization of the base k-means. We present the DRME algorithm in detail as follows with a schematic diagram of the deep architecture shown in Fig. 1.

The key step of developing a deep learning algorithm is to design f_{θ_l} for balancing the effectiveness and efficiency of the algorithm. See Section 4.2 and Ref. [1] for reviews. DRME is also a stack of building blocks $\{f_{\theta_l}\}_{l=1}^{L}$. But unlike the existing deep learning algorithms, the l-th building block f_{θ_l} is an ensemble of random models, denoted as $\left\{g_v^{(l)}\right\}_{v=1}^{V}$.

Each random model $g_v^{(l)}$ consists of the following two phases:

- **Random observation sampling and nonlinear transform.** The parameter of $g_v^{(l)}$ is k randomly selected observations from $\mathbf{Y}^{(l-1)}$, denoted as $\mathbf{M}_v^{(l)} = \left[\mathbf{m}_{v,1}^{(l)}, \ldots, \mathbf{m}_{v,k}^{(l)}\right]$, where k is a positive integer that is randomly chosen from a given range, denoted as $[k_{\min}, k_{\max}]$ with $k_{\max} \geq k_{\min} \geq 2$. Like the k-means clustering, we regard the selected k observations as k centers of the random model $g_v^{(l)}$, so that the observation $\mathbf{y}_i^{(l-1)}, \forall i = 1, \ldots, n$ is predicted as the shortest distance between $\mathbf{y}_i^{(l-1)}$ and the k centers. In this paper, the Euclidean distance is used as the metric, so that the prediction function is defined as:

$$z_{i,v}^{(l)} = \arg\min_j \left\|\mathbf{y}_i^{(l-1)} - \mathbf{m}_{v,j}^{(l)}\right\|_2^2, \forall i = 1, \ldots, n, \quad \forall j = 1, \ldots, k. \tag{2}$$

  Note that (i) the prediction via the Euclidean distance is regarded as a nonlinear transform of the input, and (ii) different random models have different k.
- **Sparse coding.** Suppose $z_{i,v}^{(l)} = j$. We extend $z_{i,v}^{(l)}$ to a k dimensional indicator vector $\mathbf{y}_{i,v}^{(l)}$, i.e. $\mathbf{y}_{i,v}^{(l)} = \left[y_{i,v,1}^{(l)}, \ldots, y_{i,v,k}^{(l)}\right]^T$, where the vector $\mathbf{y}_{i,v}^{(l)}$ takes 1 for the j-th element and 0 for the others. This 1-of-k coding method is a common strategy in multiclass problems, such as k-means.

After getting the outputs of the i-th observation from all random models $\left\{g_v^{(l)}\right\}_{v=1}^{V}$, i.e. $\left\{\mathbf{y}_{i,v}^{(l)}\right\}_{v=1}^{V}$, we concatenate these sparse vectors to a long one:

$$\mathbf{y}_i^{(l)} = \left[\mathbf{y}_{i,1}^{(l)^T}, \ldots, \mathbf{y}_{i,V}^{(l)^T}\right]^T. \tag{3}$$

Finally, we get the l-th random model ensemble $\left\{g_v^{(l)}\right\}_{v=1}^{V}$ and the l-th feature representation $\mathbf{Y}^{(l)} = \left[\mathbf{y}_1^{(l)}, \ldots, \mathbf{y}_n^{(l)}\right]$ for the $(l+1)$-th layer.

When the dimension of the sparse representation $\mathbf{Y}^{(l)}$ is much larger than the size of the data set, i.e. $d^{(l)} \gg n$, we may take the similarity matrix $\mathbf{Z}^{(l)}$ as the input of the $(l+1)$-th layer instead of $\mathbf{Y}^{(l)}$, which is a scheme we have adopted in all experiments of this paper. Given the sparse representation $\mathbf{Y}$, the similarity matrix $\mathbf{Z}$ is calculated by:

$$\mathbf{Z} = \frac{1}{n}\mathbf{Y}^T\mathbf{Y} \tag{4}$$

where $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n]$ with $\mathbf{z}_i \in [0,1]^n, \forall i = 1, \ldots, n$. The similarity matrix has been adopted in many well-known algorithms, such as [12, Section 3.2] and [13, Section 3.2]. Note that the $n \times n$ similarity matrix might be further compressed by the Nystrom method [20].

Here, we remark three items. (i) The random observation sampling is very important to the success of DRME. It is a "meaningful" base learner that is slightly better than a random guess. See Sections 4.1 and 5.3 for a further discussion. (ii) The 1-of-k coding method can be seen as one of the simplest sparse coding methods [21], see Sections 4.3 and 5.2 and [16] for a further discussion. (iii) We may further improve the performance by enlarging the diversity between the random models via the random feature selection. This topic is beyond the scope of this paper.

2.3 Complexity Analysis

For facilitating our analysis, we do not consider the difference between the layers. We make the following

notations: The depth of DRME is L. For each layer, the random model ensemble consists of V base clusterings; the average number of the output clusters of each base clustering is k; the input feature of each layer is a $d \times n$ matrix with each column representing an observation; the sparsity of the input is s, where the matrix sparsity is defined as the ratio of the number of non-zero entries to the size of the matrix.

2.3.1 Computational Complexity

It is easy to see that each base clustering g has a computational complexity of $(dnsk)$. Hence, the computational complexity of DRME is $(dnskVL)$. Because there exists the following relation:

$$d = Vk \tag{5}$$

we can conclude that the computational complexity of DRME is about $\left(nsk^2V^2L\right)$.

If we take the similarity matrix in Eq. (4) as the input of each layer, the complexity of each base clustering is about $\left(n^2sk\right)$. Calculating the similarity matrix needs an additional complexity of about $\left(n^2ds^2\right)$. Therefore, the computational complexity of DRME is about $\left(n^2skVL + n^2ds^2L\right)$. Substituting Eq. (5) to the complexity derives $\left(n^2skVL + n^2s^2kVL\right)$.

In practice, we usually set both k and V to large values, e.g. $k \approx 50$ and $V = 2000$, so as to guarantee the robustness of the performance. Hence, it is easy to observe that taking the sparse representation as the input of each layer is suitable to large scale problems, while taking the similarity matrix as the input of each layer, which is the case of this paper, is suitable to small scale problems.

2.3.2 Storage Complexity

For the l-th layer, we need to store its whole input and output, which requires an $(2dns)$ space. We also need to store $f^{(1)}, \ldots, f^{(l)}$, which requires an (kVd) space. Summing the two items equals to $(2dns + kVd)$. Substituting Eq. (5) to the summation can reach the conclusion that the storage complexity of DRME is $(2nskV + k^2V^2)$. Because k and V does not have a direct relationship with n, the overall storage complexity is linear with respect to the size of the data set.

For small scale problems, if we take the similarity matrix as the input of each layer, we need an additional storage complexity of (n^2) for the similarity matrix, which is the case of this paper.

2.4 Effectiveness of DRME: A Visualized Example

Because it is assumed in machine learning that the observations in a high-dimensional space are triggered by very few independent factors that lies in a low-dimensional subspace, the effectiveness of the learned representation is judged by whether the observations that come from different classes can be well separated in a low-dimensional embedding subspace. Hence, if we extract the low-dimensional information from the learned feature representations by e.g. PCA, a good representation can yield the following clear pattern: the observations from the same factor are concentrated, while the observations from different factors are well separated.

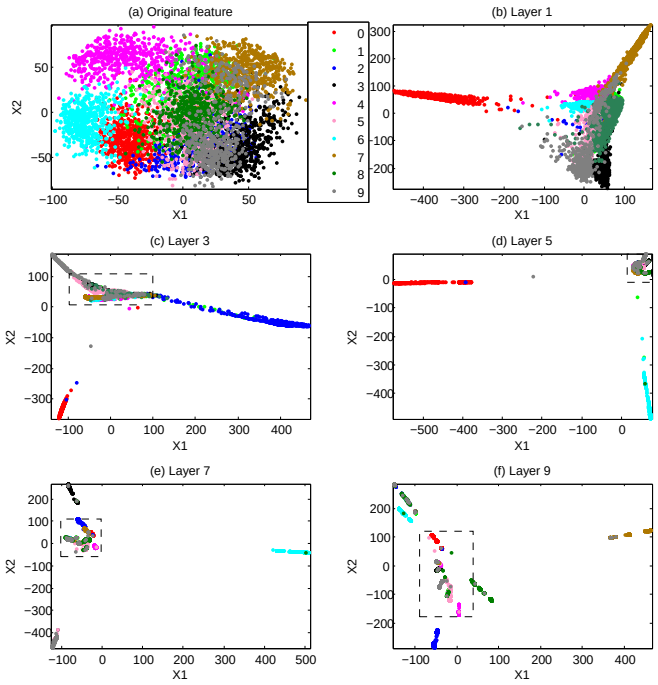

Fig. 2. Visualizations of different feature representations of the Optdigits data set. (a) Original features. (b)-(f) Learned feature representations by DRME at different layers (i.e. depths). The images are get via PCA. The images in the dashed boxes of Figs. (c), (d), (e), and (f) are further amplified in Figs. 3c, 3d, 3e, and 3f, respectively.

TABLE 1
Parameter settings of the deep random model ensemble.

Description of the parameter	Maths Notation	Value
Depth of DRME	L	10
Number of random models per layer	V	2000
Minimum number of clusters per random model	$k_{\min}$	10
Maximum number of clusters per random model	$k_{\max}$	100

In this subsection, we run DRME on the optical recognition of handwritten digits (Optdigits) data set.[1] The Optdigits data set is a widely used benchmark data set in the UCI machine learning repository. It contains 10 hand written integer digits ranging from 0 to 9. It consists of 5620 observations and 64 attributes (i.e. dimensions). Each digit consists of about 560 observations. The parameter settings of DRME are summarized in Table 1. To visualize the learned representations, we project them to a 2-dimensional subspace by PCA.

The result is shown in Fig. 2. The images in the dashed

1. http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

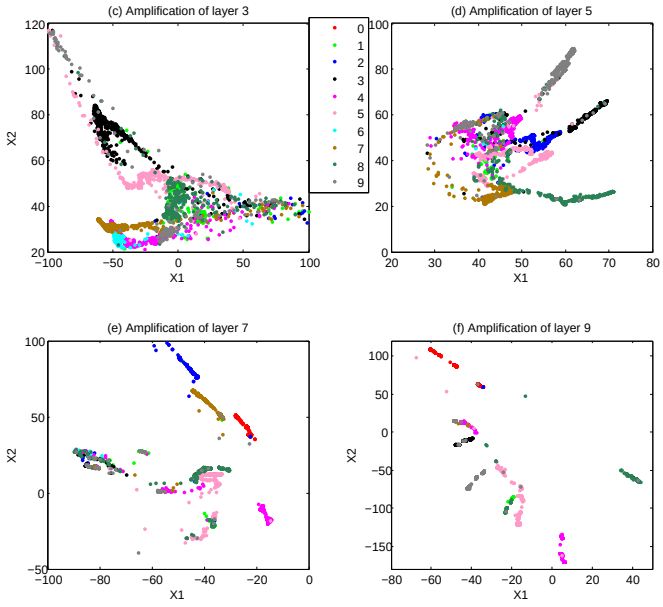

Fig. 3. Amplifications of the images in the dashed boxes of Fig. 2. Figs. (c), (d), (e), and (f) are the amplifications of Figs. 2c, 2d, 2e, and 2f, respectively.

boxes of Figs. 2c, 2d, 2e, and 2f are amplified in Figs. 3c, 3d, 3e, and 3f, respectively. From the figures, we can see clearly that when the depth of DRME increases, the observations from the same digit are becoming more and more concentrated while the observations from different digits are becoming more and more separated, which fully meets our expectation.

3 Deep Random Model Ensemble for Clustering

In this section, we will first present the importance of applying DRME to clustering. Then, we will present the DRME based clustering in detail. At last, we will illustrate the effectiveness of the DRME based clustering with one visualized example.

3.1 Motivation

Applying DRME to clustering has two important goals: (i) detecting the most powerful representation among the layers, and (ii) detecting the natural clusters. The first goal is the main objective of this application, while the second one can be regarded as an important by-product of the application.

3.1.1 Why Do We Use Clustering to Detect the Most Powerful Representation?

When the depth is chosen properly, the learned representation is close to the underlying smooth manifold. However, when the depth of DRME is not deep enough, we may not get a smooth feature representation, which is known as *under-fitting*. Also, when the data set is not large scale, we get the risk of *over-fitting*. Hence, it is important to decide which representation we should pick up among the layers.

If we have no knowledge about the data, clustering seems the only way for this problem. Moreover, compared to the supervised learning, the performance of clustering is much more sensitive to the shape of the representation, hence, using clustering to detect the changes of the representations among the layers is better than using supervised learning.

3.1.2 Why Is This Application Important to Clustering?

Clustering is the process of partitioning a set of data observations into multiple clusters so that the observations within a cluster are similar, and the observations in different clusters are very dissimilar [22]. Data representation is the core problem of clustering. Specifically, as summarized in [23, Section 3], data clustering has four challenges, which are the (i) data representation, (ii) purpose of grouping, (iii) number of clusters, and (iv) cluster validity. Among the challenges, data representation is the base of the other three. First, different purpose of grouping needs different data representations. Second, as shown in [23, Fig. 5] and Fig. 2, a good representation that reflects the essential factors can result in a clear data structure, so that a simple clustering algorithm can reach a valid partition.

3.2 DRME Based Clustering

Any clustering algorithm that is able to yield unfixed number of clusters can be combined with DRME. Generally, clustering algorithms can be divided into two groups: *partitional* and *hierarchical*, see [23], [24] for excellent reviews. Although some partitional algorithms can yield unfixed number of clusters, such as Dirichlet process mixture models [25] or support vector clustering [26], in this paper, we prefer the representative hierarchical clustering methods, such as single-linkage or complete-linkage, since they are simple, fast, and need no parameter tuning.

In this paper, the *single-linkage clustering* is used. It is an agglomerative hierarchical clustering method. Specifically, it builds a hierarchical-tree on the data. Each merging of two leaves (i.e. clusters) generate a new partition of the data. If we record the distances between the merged leaves, the tree can be presented as a vector with $n-1$ elements (i.e. distance records), denoted as $\mathbf{p} = [p_1, \ldots, p_{n-1}]^T$ with p_1 and p_{n-1} as the last and first mergings respectively. Note that $\mathbf{p}$ is in the descend order with p_1 as its largest value.

As [13, Section 3.3] did, the number of clusters $k^\star$ is selected as the one that yields the longest cluster lifetime [13, Fig. 3]:

$$k^\star = \arg\max_k p_{k-1} - p_k, \quad \forall k = 2, \ldots, n-1. \tag{6}$$

However, because a manually-defined class might have several natural clusters, selecting a good representation according to the number of clusters only might be too

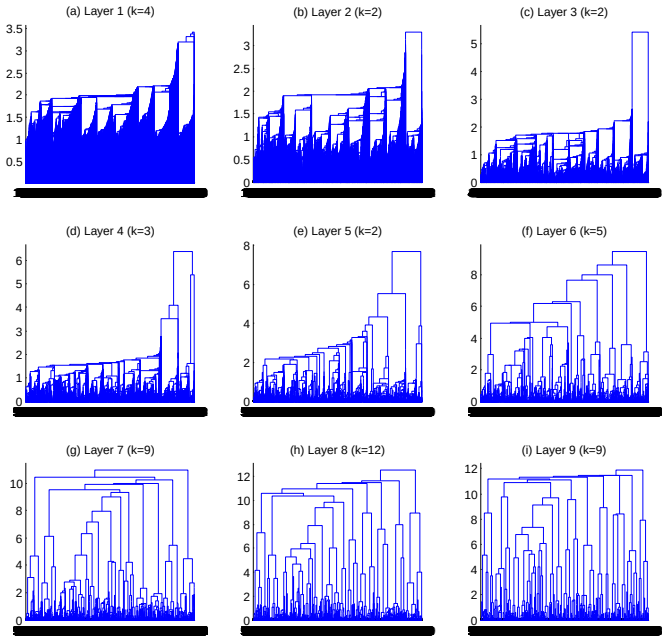

Fig. 4. Dendrograms produced by the single-linkage on the Optdigits data set. k in the title of each dendrogram is the number of the detected clusters.

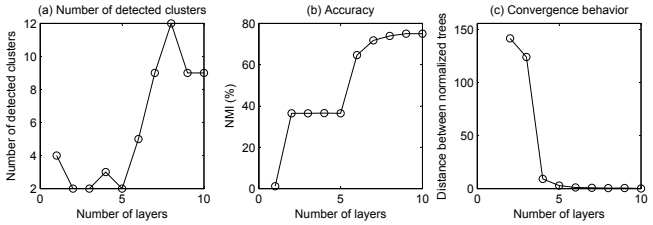

Fig. 5. Experimental results of the DRME based clustering on the Optdigits data set. (a) Curve of the detected cluster numbers. (b) Accuracy curve. (c) Curve of the distances between the successive normalized hierarchical-trees.

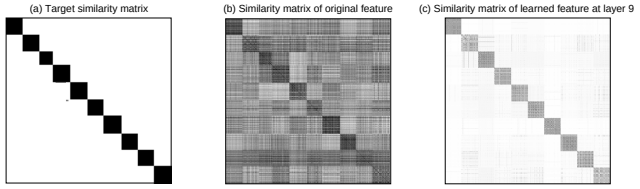

Fig. 6. Similarity matrix comparison on the Optdigits data set. (a) Target similarity matrix calculated from the ground truth labels. (b) Similarity matrix on the original features. (c) Similarity matrix on the learned feature representation at layer 9.

arbitrary. Moreover, it is not robust: a slight disturbance on the representation might yield a very different $k^\star$, so that it is hard to design a simple criterion that is based on the longest cluster lifetime for the representation selection problem.

In this paper, **we propose to select the robust feature representations according to the distance between the normalized hierarchical-trees of the successive two layers.** Specifically, given the trees of all layers $\{\mathbf{p}^{(l)}\}_{l=1}^{L}$, we first normalize the tree of each layer by $\hat{\mathbf{p}}^{(l)} = \mathbf{p}^{(l)}/p_1^{(l)}$, and then calculate the distance between the successive two normalized trees as:

$$q_l = \left\| \hat{\mathbf{p}}^{(l)} - \hat{\mathbf{p}}^{(l-1)} \right\|_2^2, \quad \forall l = 2, \ldots, L \tag{7}$$

At last, we pick up the output of the *first* layer that satisfies the following inequality as the learned representation:

$$\frac{|q_{l^\star} - q_{l^\star - 1}|}{\left|\max_{l=2}^{L} q_l - q_{l-1}\right|} \leq \eta \tag{8}$$

where $l^\star$ represents the layer, and $\eta \in (0, 1)$ is a user defined constant. Note that this criterion is only an empirical one. The monotonic decrease of $q_2, q_3, \ldots, q_L$ is unguaranteed.

3.3 Effectiveness of the DRME Based Clustering: A Visualized Example

In this subsection, we will run the DRME based clustering on the Optidigits data set. The accuracy of the proposed clustering algorithm is evaluated as comparing the predicted labels with the ground truth labels using normalized mutual information (NMI), where NMI was proposed in [12, Eq. (3)] and has been one of the standard metrics for clustering.

The dendrograms produced by the single-linkage are shown in Fig. 4. From the figures, we can see clearly that the proposed DRME has a strong denoising ability.

The experimental results are summarized in Fig. 5. From the figure, we can observe that (i) the unstable variation of the detected clusters does not affect the clustering accuracy much, (ii) the clustering accuracy is mainly determined by the learned representation, and (iii) rather than taking the number of the detected clusters as the representation selection criterion, using the distance between the successive normalized hierarchical-trees as the selection criterion is a good choice.

Note that the k-means clustering provided with the true cluster number can only achieve an NMI of 72.93% while the DRME based clustering can achieve 75.00%, which further demonstrates the power of the deep representation learning algorithm.

4 Related Work

In this section, we will briefly present three related topics which are clustering ensemble, deep learning, and sparse coding, respectively.

4.1 Clustering Ensemble

Before reviewing clustering ensemble, we should first review the supervised ensemble learning where the clustering ensemble is rooted in.

Ensemble learning aims to combine a group of diverse base learners together for a better performance. The success of the ensemble methods relies heavily on the following two basic criteria [11].

- A *meaningful* selection of the base learner. The keyword "meaningful" means that the base learner needs to be at least better than a random guess.
- A strong *diversity* among the base learners. The keyword "diversity" means that when the base learners make predictions on an identical pattern, they are different from each other in terms of errors.

As presented in [11], there are generally four groups of ensemble learning methods, which are the methods of manipulating the training examples [27], manipulating the input features [28], manipulating the training parameters [29], and manipulating the output targets [30].

Clustering ensemble is an extension of the ensemble learning to unsupervised learning [12]–[15]. The key advantage of the clustering ensemble to single clustering is that a group of clusterings are capable of grasping the shape of highly variant data and have the potential of preventing bad local minima that most clustering algorithms suffer from. Typically, a clustering ensemble is broken into two components: (i) a group of clusterings that yield different partitions, and (ii) a *consensus function* that aims to combine the partitions (i.e. the base clusterings). As its supervised counterpart, clustering ensemble should satisfy the aforementioned two key criteria, and can construct diverse base clusterings in the aforementioned four ways. Currently, researchers focus on designing the consensus function which is a self-contained problem of clustering ensemble. See [14] for an excellent review of the consensus function.

But if we regard the group of different partitions as a new feature representation of the original data, and if we regard the consensus function as a clustering algorithm running on the new representation, the clustering ensemble problem is reduced to a single clustering problem applied on the output of one layer (probably nonlinear) transform of the original data. Because, as has been summarized in [23], [24], the clustering performance is mostly decided by the shape (i.e. the feature representation) of the data but not the clustering algorithm, it might be better for us to pay more attention to the unsupervised feature representation learning subproblem.

Clustering ensemble contributes to the key motivation of this paper. First, it motivates us to view the representative deep learning approaches as a stack of clustering ensembles. Second, the two basic criteria of ensemble learning contributes to the guidance of our design of the proposed DRME. Third, the four types of diversity enhancement techniques contribute to the implementation skill of our random model ensemble in each layer. Fourth, the clustering ensemble problem provides us a good testing environment about whether the learned feature representation can reveal the underlying natures of the data.

4.2 Deep Learning

In [1], Bengio *et al.* have conducted an excellent review on deep learning and representation learning. Here, we briefly summarize part of its content that is related to this paper.

Existing deep learning approaches can be categorized to two classes, which are rooted in *probabilistic graphic models* and *neural networks* respectively [1, Section 5]. The main difference between them are how to interpret the hidden units: latent random variables in probabilistic graphic models or computational nodes in neural networks?

The representative method rooted in probabilistic graphic models is the deep belief networks (DBN) [2]. Its building block is RBM, which is a typical kind of undirected graphic models that lies in the exponential families. The main merit of RBM to the popular directed graphic models is that the conditional distribution over the hidden units can factorize given the visible units, and vise versa, so that most inferences are readily tractable [1, Section 6.2.1]. The objective of RBM is to maximize the likelihood of the input. It is solved by the stochastic gradient descent algorithm. The detailed derivation of the algorithm can be found in [3]. The main difficulty of the optimization is that the expectation of the partition function (the normalization term) of the probabilistic model is still computationally untractable, so that the expensive Markov chain Monte Carlo (MCMC) sampling has to be used for this function. Surprisingly, in [17], [18], Hinton found that it is needless to carry out the full MCMC, conducting only few steps of MCMC can also achieve good results. This biased approximation of maximum likelihood learning is named *contrastive divergence* (CD) learning. In [17], [18] and [1, Section 9.4], Hinton and Bengio *et al.* have tried to explain why the CD learning can provide a reasonable approximation of the maximum likelihood learning.

The representative method rooted in neural networks is the stacked denoising autoencoder (SDAE) [6], [7]. Its building block is DAE, which is a regularized autoencoder. Compared to the probabilistic graphic model based approaches, the main merit of DAE is that it not only defines a simple tractable optimization objective that prevents dealing with the complicated partition function, but also can take the output of the autoencoder as the learned representation directly [1, Section 7]. Compared to the non-regularized autoencoder, the main merit of DAE is that it can learn *over-complete* representations, i.e. $y^{(l)} \geq y^{(l-1)}$, and meanwhile prevent learning nothing but duplicating the inputs [1, Section 7.2]. The objective of DAE is to minimize the reconstruction error, which is formulated as the following optimization problem:

$$\min_{\theta} \sum_{i=1}^{n} \ell(\mathbf{x}_i, h_\theta(f_\theta(\hat{\mathbf{x}}_i))) \tag{9}$$

where $\hat{\mathbf{x}}$ is a noise-corrupted version of $\mathbf{x}$, f_θ is the

encoder, h_θ is the *decoder* that will be discarded in the final network, and ℓ is the risk function. $\ell(\mathbf{x}, \mathbf{y})$ can be defined as the squared loss $\|\mathbf{x} - \mathbf{y}\|^2$ for unbounded real-valued $\mathbf{x}$ and $\mathbf{y}$, or the binary cross-entropy loss $-\sum_{t=1}^{d} x_t \log(y_t) + (1 - y_t)\log(1 - x_t)$ for $\mathbf{x}$ and $\mathbf{y}$ that are bounded in the range $[0, 1]$.

The two representative deep learning approaches contribute to two ideas of this paper. First, different from the above two branches, this paper view the existing deep learning approaches in a new perspective – knowledge reuse algorithms of clustering ensemble, where the building blocks, such as RBM and DAE, can be interpreted as clustering ensemble approaches, see Section 5.1 for a detailed discussion. Second, the interesting CD learning contributes partially to the idea of reducing the k-means ensemble [13] to the random model ensemble, see Section 5.3 for a detailed discussion.

4.3 Sparse Coding and Dictionary Learning

Sparse coding is an important unsupervised representation learning approach. It has been widely used in computer vision and image processing, and is an active subfield of machine learning. Suppose we are to learn a $d_{\mathbf{y}}$-dimensional sparse representation of the input $\mathbf{x}$, denoted as $\mathbf{y}$. The basic problem of sparse coding is formulated as the following optimization problem:

$$\min_{\{\mathbf{y}_i\}_{i=1}^n, \mathbf{M}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{M}\mathbf{y}_i\|_2^2 + \lambda \|\mathbf{y}_i\|_1, \quad (10)$$
$$\text{subject to } \|\mathbf{M}_{:,j}\|_2^2 = 1, \quad \forall j = 1, \ldots, d_{\mathbf{y}}$$

where $\mathbf{M}$ is a $d \times d_{\mathbf{y}}$ matrix variable, called the *dictionary* or the *basis set*, with $\mathbf{M}_{:,j}$ representing the j-th element of the dictionary, called the *basis vector*, and λ is a user defined parameter. The sparsity of $\mathbf{y}$ is enforced by the l_1-norm penalty. Typically, the alternating optimization method is adopted [31]. The method iterates the following two steps. The first step is to optimize $\mathbf{y}$ given fixed $\mathbf{M}$, and the second step is to optimize $\mathbf{M}$ given fixed $\mathbf{y}$. The first step can be viewed as an encoding method that can be studied and applied independently. The second step is also named *dictionary learning*.

There are many sparse coding and dictionary learning algorithms. In this paper, we pay particular attention to [16]. In [16], Coates and Ng conducted a broad experimental comparison on sparse coding, and drew two important experimental conclusions:

- *"When using sparse coding as the encoder, virtually any training algorithm can be used to create a suitable dictionary."*
- *"Regardness of the choice of dictionary, a very simple encoder can often be competitive with sparse coding."*

In this paper, we provide a reasonable explanation to the experimental phenomena of [16] in the perspective of clustering ensemble, and also view the proposed algorithm in the sparse coding perspective, see Section 5.2 for a detailed discussion. The experimental phenomena on sparse coding provide a confidential evidence to the correctness of the proposed DRME, see Section 5.3 for a detailed discussion.

5 Motivation

In this section, we will first explain why we can use the random model ensemble as the building block of deep learning by analyzing several existing representation learning approaches in Sections 5.1, 5.2, and 5.3, and then analyze empirically the key elements that contribute to the success of DRME in Section 5.3.

5.1 Viewing Deep Learning As A Framework of Knowledge Reuse of Clustering Ensemble

As presented in Section 1, deep learning is a stack of shallow models, where each shallow model takes the output of its ancestor model as its input. Hence, deep learning is a framework of knowledge reuse of shallow models. In this subsection, we focus on discussing the relationship between the representative shallow models (i.e. RBM and DAE) and clustering ensemble.

5.1.1 Relationship Between Restricted Boltzman Machine and Clustering Ensemble

RBM is a probabilistic clustering ensemble with each base clustering as a binary-class probabilistic clustering. We present this relationship in detail as follows:

The central problem of unsupervised representation learning is to model complicated smooth distributions arbitrarily accurately. As summarized in [17, Section 1], the data modeling is categorized to two classes – *mixture model* and *product of experts*.

The first class is the mixture model. It aims to combine a large number of tractable probabilistic models by forming a weighted mixture. The general probability framework of this class is as follows:

$$p_{\theta_1,\ldots,\theta_k}(\mathbf{x}) = \sum_{i=1}^{k} \pi_i p_{\theta_i}(\mathbf{x}), \quad \text{subject to} \sum_{i=1}^{k} \pi_i = 1, \quad (11)$$

where $k \in \{1, 2, \ldots, +\infty\}$ is the number of the mixtures, π_i is the weight of the i-th individual model, and p_{θ_i} is the i-th probabilistic model with θ_i as its parameter. One typical model of this class is the Gaussian mixture model (GMM). It is well-known that GMM is a probabilistic clustering algorithm with each mixture modeling a cluster. Another typical model is the k-means clustering, which is a small-variance asymptotics (i.e. a hard clustering version, or deterministic version) of GMM [32, Chapter 9.3.2]. These models are easily optimized via the EM algorithm. However, they are ineffective in modeling the posterior distributions that are sharper than the individual mixtures.

The second class is the product of experts. It aims to combine multiple individual models by multiplying them, where the individual models have to be a bit more complicated and each contains one or more latent

variables. The general probability framework of this class is as follows:

$$p_{\theta_1,\ldots,\theta_k}(\mathbf{x}) = \frac{\prod_{i=1}^{k} f_{\theta_i}(\mathbf{x})}{\sum_{\mathbf{x}'}\prod_{i=1}^{k} f_{\theta_i}(\mathbf{x}')} \tag{12}$$

where $\mathbf{x}'$ indexes all possible vectors in the data space, and f_{θ_i} is called an *expert* [17]. One typical model of the second class is the Bernoulli-Bernoulli RBM model. Its expert f_{θ_i} is specified as:

$$f_{\theta_i}(\mathbf{x}) = \sum_{h_i\in\{0,1\}} e^{c_i h_i + h_i \mathbf{W}_{i,:}\mathbf{x}} \tag{13}$$

where $h_i \in \{0,1\}$ is a binary hidden variable, and $\theta_i = \{c_i, \mathbf{W}_{i,:}\}$ with $\mathbf{W}_{i,:}$ as the i-th row of parametric matrix $\mathbf{W}$ of RBM and c_i as the i-th bias term. See [3] for a detailed derivation of Eq. (13). The difficulty of this class is that the denominator of Eq. (12) is untractable, so that expensive MCMC has to be used. If we roughly view (13) as a binary-class probabilistic clustering, RBM is in fact a probabilistic clustering ensemble with each base clustering shown in (13). More generally, regardless of the difficulty of the parameter inference, we can substitute (11) to (12) for any complicated clustering ensemble model. Hence, it is not surprising that the product of experts can achieve superior performance than the mixture model in many applications, such as the RBM based speech recognition system (without a deep structure) over the GMM based one [33].

5.1.2 Relationship Between Denoising Autoencoder and Clustering Ensemble

Both DAE and RBM belong to the class of product of experts, i.e. clustering ensembles. The difference between them is that DAE is a deterministic clustering ensemble while RBM is a stochastic one. This difference is analogous to the difference between GMM and k-means in the class of mixture model.

Specifically, DAE is an ensemble of the following binary-class deterministic clustering:

$$f_{\theta_i}(\mathbf{x}) = \frac{1}{1+e^{-c_i-\mathbf{W}_{i,:}\hat{\mathbf{x}}}} \tag{14}$$

where $\theta_i = \{c_i, \mathbf{W}_{i,:}\}$ is the classification hyperplane of the clustering, $\hat{\mathbf{x}}$ is a random feature sampling of $\mathbf{x}$. Note that the random feature sampling is one of the most important diversity enhancement techniques of ensemble learning [28].

Eventually, it is valid to use any clustering ensemble whose base learner satisfies the two criteria in Section 4.1 as the building block of a deep architecture. In this paper, we propose to use the framework in [13] as the building block. This building block has the following two properties:

- The base learner is a multi-class clustering that can partition data to arbitrary number of clusters.
- The output of the base learner is a 1-of-k sparse representation.

Besides, we may take randomly selected features as the input of the base learner as [15] and DAE did, though we have observed no obvious performance improvement on the experimental data sets when adopting this scheme.

5.2 Viewing Sparse Coding As the Consensus Function of Clustering Ensemble

In this subsection, we will focus on analyzing two interesting experimental phenomena of [16], which is summarized in Section 4.3, in the perspective of clustering ensemble. The main conclusion of this analysis is that **when sparse coding is used as the encoder, it is equivalent to the consensus function of clustering ensemble, and dictionary learning is equivalent to the training process of the base learners.**

Specifically, given a learned dictionary $\mathbf{M} \in [-1,1]^{d\times d_{\mathbf{y}}}$, sparse coding aims to solve the following optimization problem:

$$\min_{\mathbf{y}_i} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{M}\mathbf{y}_i\|_2^2 + \lambda\|\mathbf{y}_i\|_1 \tag{15}$$

It is known that the parameter λ controls the sparsity. Here, we view it in a different way – a parameter that controls the number of the base learners of the clustering ensemble. Specifically, if we set $\lambda = 0$, it is likely that $\mathbf{y}_i$ contains only one non-zero element. That is to say, we group all basis vectors to a single $d_{\mathbf{y}}$-class clustering, which is obviously a weak consensus function. A good value of λ is the one that can make a small part of the elements non-zero. This choice is equivalent to partitioning the dictionary to several (probably overlapped) subsets and then grouping the basis vectors in each subset to a base clustering. From this point of view, only if the base learners satisfy the two basic criteria, no matter how weak the base learners are and what kind of sparse coding is used, the performance of sparse coding (as an encoder) is guaranteed. This accounts for the experimental phenomena of [16].

In fact, **the random model ensemble in the proposed DRME is one of the simplest sparse coding method, and is implemented in a similar way with what we have analyzed above.** Specifically, we first randomly sample multiple observations (for example, 500) to form a dictionary; then, we randomly partition the dictionary to a serial highly overlapped subsets (for example, 2000 subsets) with each subset containing an arbitrary number of basis vectors within a given range (for example, $[10, 100]$); finally, we regard each subset as a base clustering and adopt the 1-of-k coding, which is the simplest sparse coding method, to each base clustering. Note that why we can use the overlapped subsets can be explained by the *explaining away* property of sparse coding, see [1, Sections 6.1.1 and 6.1.3] for a detailed analysis.

Moreover, we can also explain two important experimental phenomena of [16], which is not emphasized in [16], in the perspective of clustering ensemble. (i) The

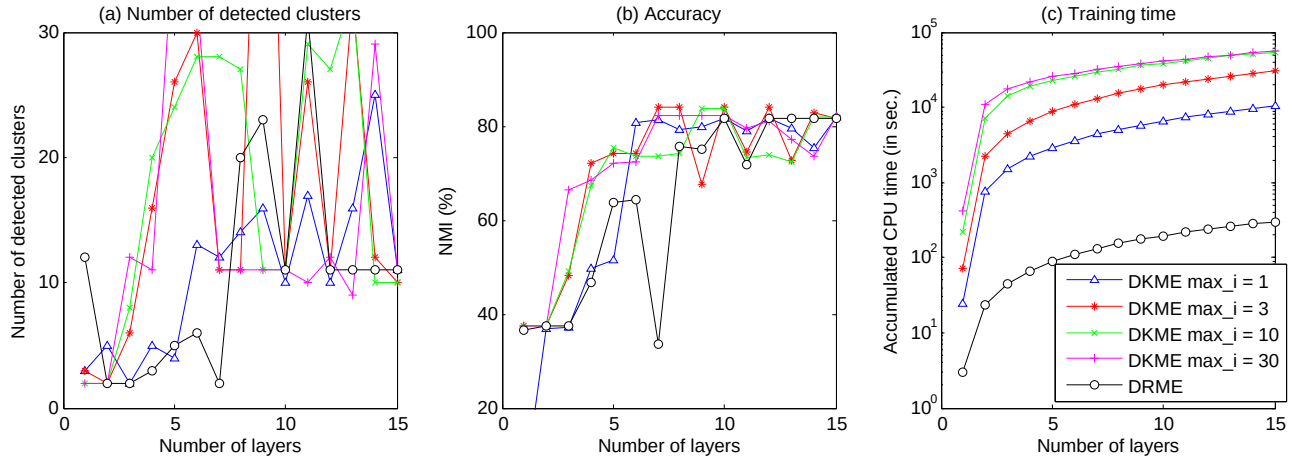

Fig. 7. Results of the reduction of the EM iterations on the Optdigits data set. "DKME" is short for deep k-means ensemble. "max_i" represents the maximum iteration number of the base k-means.

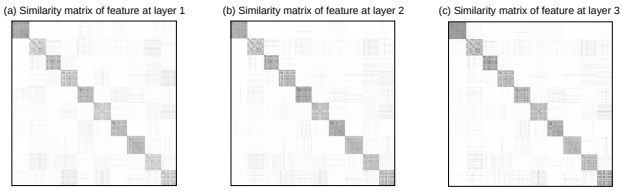

Fig. 8. Similarity matrices of the first three layers of DRME on the Optdigits data set with the base clustering generated from the random observation sampling.

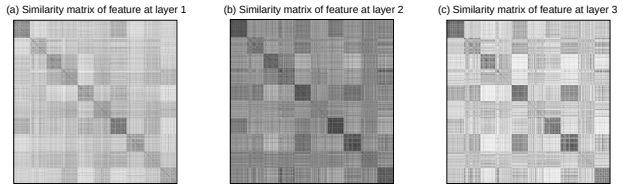

Fig. 9. Similarity matrices of the first three layers of DRME on the Optdigits data set with the base clustering generated from the completely random centers.

only failed dictionary in [16] is the one that consists of completely random weights. The reason is that this dictionary does not satisfy the first basic criterion of ensemble learning. In other words, the base learners are too weak to be meaningful ones. (ii) The dictionary that is filled via random sampling from the input observations, which is a scheme coincident with the proposed DRME, can achieve the state-of-the-art performance. This is because random sampling is not completely random, it can reflect the distribution information of the input and hence is probably the weakest meaningful base learner we can find.

5.3 Replacing Clustering Ensemble With Random Model Ensemble

As mentioned in Section 5.1, we have adopted the framework of the k-means based clustering ensemble in [13]. After viewing RBM, DAE and sparse coding as special cases of clustering ensemble and revisiting the basic criteria of clustering ensemble, we are ready to reduce the k-means based clustering ensemble that is trained with the full EM training to the one that is trained with only one or even zero EM iteration. This reduction is mainly motivated from the biased approximation of the CD learning to maximum likelihood learning for DBN [17], [18], and is further supported by the success of the random sampling based dictionary learning when sparse coding is used as the encoder [16].

Fig. 7 illustrates the effectiveness of this reduction on the Optdigits. In this example, we use a subset of Optdigits that consists of only 2000 observations, and use only 200 base clusterings per layer. From the figure, we can observe that (i) even if we use a light experimental setting, the deep k-means ensemble (DKME) is still very inefficient, even with only one EM iteration, while DRME is about two orders faster than the DKME with only one EM iteration; (ii) although DKME can reach good representations with less layers than DRME, both of them can reach equivalently good representations in very deep layers.

Here comes the question. Can we use completely random centers instead of the centers that are randomly sampled from the data? No, from Fig. 9, we can see that when we use completely random centers, the similarity matrices are quite confused, while from Fig. 8, we can observe that when we use the random observation sampling, the similarity matrices are getting clearer with the increase of the depth.

As a conclusion, **the random observation sampling is empirically a meaningful base clustering for the deep clustering ensemble.**

6 Experiments

In this section, we will compare the proposed DRME algorithm with 5 referenced representation learning algorithms on 19 UCI benchmark data sets. All experiments

TABLE 2
Descriptions of the data sets. The data sets that are marked with * are the randomly sampled subsets from the original data sets.

ID	Data	Size (n)	Feature (d)	Class (k)
1	Dermathology	366	34	6
2	Iris	150	4	3
3	Ecoli	336	4	3
4	Wine	178	7	8
5	Glass	214	9	6
6	New-Thyroid	215	5	3
7	Vowel	990	10	11
8	Balance	625	4	3
9	Yeast	1484	8	10
10	Satimage*	2000	36	6
11	Letter*	2000	16	26
12	Pendigits*	2000	16	10
13	Segmentation*	2000	19	7
14	Optdigits*	2000	64	10
15	Shuttle*	2000	9	5
16	Vehicle	846	18	4
17	Fea*	2000	87	5
18	Libras	168	90	7
19	Synthetic-Control	600	60	6

are conducted with MATLAB 7.12 on a 2.27 GHZ 8-core Itel(R) Xeon(R) Server running Windows XP with 16 GB memory. The implementation of DRME can be downloaded from http://XXXXX.

6.1 Experimental Settings and Comparison Schemes

The experiments are performed on 19 UCI data sets.[2] In this paper, we conduct 20 independent runs on each dataset and report the average results. For the original UCI data sets that are more than 2000 observations, we randomly sample 2000 observations for 20 times and conduct each independent run on different samplings. The detailed information of the data sets are listed in Table 2.

For the proposed DRME, the depth is set to 15. The size of the dictionary is set to 500. The number of the base clusterings in each layer is set to 2000. The minimal number of clusters that the base clustering can achieve, i.e. $k_{\min}$, is set to 10. When the size of the data set is smaller than 500, the maximal number of clusters that the base clusterings can achieve, i.e. $k_{\max}$, is set to 30, otherwise, $k_{\max}$ is set to 100.

To examine the effectiveness of the proposed DRME, we compare DRME with the following 5 representation learning methods.

1) Shallow representation learning methods.
   - k-means based clustering ensemble (KMCE) [13]. The number of the base clusterings is set to 2000. According to [13], $k_{\min}$ is set to 10. $k_{\max}$ is set to 30.
   - Random model ensemble (RME). This is the DRME with a depth of only 1 layer. The other parameters are set to the same values as the proposed DRME.
   - Principle component analysis (PCA). The kernel PCA [34] toolbox[3] is used with the kernel type set to the linear kernel. The largest 100 eigenvalues corresponding with their eigenvectors are preserved.
2) Deep representation learning methods.[4]
   - Deep belief networks (DBN) [2]. The depth is set to 5. The number of the hidden units in each layer is set to 200. The learning rate is set to 0.005. The momentum is set to 0.9. The number of epoches for the unsupervised training is set to 120. The batch size of observations is set to 1.
   - Stacked denoising autoencoder (SDAE) [6], [7]. The depth is set to 5. The number of the hidden units in each layer is set to 200. The learning rate is set to 0.005. The fraction of the zero-masked inputs is set to 0.5.

Note that the reason why we set the depth to 5 but not 15 (as we did in DRME) is because we found that the performance of DBN and SDAE drops significantly when the depth is extremely deep.

For each representation learning method, the effectiveness of the learned representation is evaluated by the accuracy and number of clusters yielded from the single-linkage, where the accuracy is evaluated by NMI [12, Eq. (3)]. For the DRME-based single-linkage clustering, the parameter η is set to 0.005. We will also report the highest NMI that the DRME-based single-linkage can achieve among the layers. The corresponding ideal method is denoted as iDRME. For the DBN-based and SDAE-based single-linkages, we pick up the highest NMIs they can achieve among all 5 layers. Note that this is an unfair comparison scheme to our DRME, but we dare to compare in this way.

Besides the aforementioned representation learning methods, we will further provide the performance of the k-means[5] provided with the true number of clusters. The corresponding method is denoted as KM*.

For all 6 representation learning methods, only the CPU time that is consumed on learning the representations is recorded.

6.2 Results

In this subsection, we will compare the clustering accuracy in terms of NMI, the detected number of clusters,

2. http://archive.ics.uci.edu/ml

3. The implementation code is in the SVM-KM toolbox "http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/index.html".

4. The deep learning toolbox is downloaded from "https://github.com/rasmusbergpalm/DeepLearnToolbox".

5. The implementation code is in the VOICEBOX developed by Cambridge University for speech processing. It can be downloaded from "http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/doc/voicebox/kmeans.html"

TABLE 3
NMI (in percentage) comparison. The digit in brackets is the standard deviation. The digit in italic and red color means that the corresponding method is over-fitting to the data set, hence it is meaningless. The digit in bold means that the corresponding method achieves the highest NMI on the data set. We test for confidence interval at 95% with the two-tailed t test. The column of "KM*" lists the NMIs of the k-means provided with the true number of clusters. The column of "iDRME" lists the highest NMIs that the DRME can achieve. These two columns will not join in the comparison.

ID	Data	KM*	KMCE	RME	PCA	DBN	SDAE	DRME	iDRME
1	Dermathology	83.87 (4.98)	**93.68 (0.00)**	54.23 (0.00)	54.23 (0.00)	83.06 (3.05)	54.93 (5.35)	82.57 (5.66)	86.61 (2.44)
2	Iris	71.76 (4.41)	65.70 (0.00)	**76.12** (0.00)	**76.12** (0.00)	70.81 (11.34)	**76.12** (0.00)	68.62 (3.29)	76.12 (0.00)
3	Ecoli	59.16 (2.35)	29.98 (7.69)	**55.86** (0.15)	20.57 (0.00)	32.03 (9.09)	21.40 (0.84)	**54.62** (3.06)	61.75 (3.04)
4	Wine	83.08 (1.75)	49.56 (4.72)	*67.50* (12.70)	2.67 (0.00)	23.48 (10.73)	13.86 (16.13)	**66.60** (9.14)	73.69 (3.94)
5	Glass	32.33 (3.95)	27.71 (9.59)	*39.97* (9.13)	8.98 (0.00)	18.20 (9.69)	10.31 (1.70)	**37.35** (1.62)	*43.51* (6.62)
6	New-Thyroid	60.27 (0.00)	27.63 (0.00)	17.41 (19.89)	8.93 (0.00)	18.27 (8.94)	14.94 (11.05)	**47.56** (1.72)	51.89 (3.39)
7	Vowel	38.13 (2.44)	10.53 (0.00)	7.44 (3.30)	13.64 (0.00)	*12.81* (9.82)	12.77 (2.12)	**39.22** (10.29)	46.50 (3.30)
8	Balance	11.76 (6.63)	*29.45* (0.00)	*20.35* (17.30)	3.92 (0.00)	4.39 (0.76)	*37.67* (0.02)	**13.58** (4.82)	*33.24* (4.71)
9	Yeast	27.68 (1.00)	11.45 (0.00)	*26.62* (22.26)	6.34 (1.17)	9.31 (2.35)	7.72 (1.10)	**22.93** (6.55)	*38.57* (10.60)
10	Satimage*	61.79 (0.80)	13.63 (15.72)	*3.88* (10.03)	1.96 (0.77)	31.92 (7.90)	8.43 (11.95)	**48.76** (11.34)	58.94 (1.55)
11	Letter*	38.66 (1.55)	10.12 (0.82)	*15.40* (21.39)	*33.65* (31.83)	*20.13* (23.50)	*55.89* (23.42)	**17.74** (7.50)	*41.38* (10.61)
12	Pendigits*	67.64 (2.14)	22.83 (9.11)	15.00 (18.58)	2.66 (0.99)	33.67 (7.53)	3.98 (1.86)	**60.74** (15.94)	75.76 (2.30)
13	Segmentation*	61.54 (1.61)	**63.26** (0.32)	**63.18** (0.69)	49.92 (8.05)	45.89 (0.35)	49.82 (23.90)	**60.44** (10.40)	66.14 (1.73)
14	Optdigits*	73.17 (2.85)	40.16 (7.68)	18.27 (18.34)	1.56 (0.48)	20.60 (18.90)	*31.15* (27.09)	**67.34** (15.07)	82.04 (2.09)
15	Shuttle*	37.02 (3.94)	**46.26** (10.69)	**45.65** (13.26)	29.44 (25.38)	8.30 (9.49)	30.49 (21.39)	37.11 (8.24)	56.65 (8.17)
16	Vehicle	10.98 (2.06)	10.13 (0.00)	*12.55* (16.48)	1.43 (0.00)	14.23 (3.12)	6.58 (9.13)	**18.63** (3.65)	*29.02* (8.46)
17	Fea*	15.81 (8.67)	5.73 (7.18)	6.47 (3.84)	4.16 (2.61)	7.80 (11.31)	*39.22* (0.46)	**12.47** (10.07)	28.90 (3.05)
18	Libras	48.22 (4.55)	13.66 (0.00)	13.66 (0.00)	4.39 (0.00)	21.33 (12.97)	31.12 (23.31)	**60.62** (4.66)	66.39 (0.73)
19	Synthetic-Control	72.78 (2.42)	**82.71** (0.00)	50.15 (0.00)	50.15 (0.00)	65.02 (8.95)	50.15 (0.00)	64.34 (20.59)	82.23 (1.64)

and the CPU time, respectively.

Tables 3 and 4 list the clustering accuracy and the detected number of clusters respectively. We should consider the two tables jointly. Before our formal analysis, we have to note that when the detected number of clusters is very high, the clustering accuracy is generally high, however, this is an illusion since the single-linkage fails to detect useful clusters. This is mostly caused by the roughly learned representation. Therefore, in our comparison, when the detected numbers of clusters in Table 4 are very high, we will not consider the corresponding results both in Table 4 and in Table 3 anymore. From the two tables, we can observe the following experimental phenomena. (i) The proposed DRME can achieve the highest NMIs and detect the true numbers of clusters in most of the 19 data sets. (ii) Generally, DRME achieves an accuracy as high as KM*, and moreover, iDRME is even better than KM*. (iii) The proposed representation selection scheme works quite well, while the referenced methods suffer more or less from the under-fitting problem including iDRME. (iv) The referenced deep learning approaches does not achieve the expected performance. One reason might be that the parameters are not well-tuned. However, we have no way to tune the parameters in the real-world unsupervised learning scenario, hence, only the empirically workable settings are adopted. Another reason is that the data is so small scale that it cannot meet the requirement of the parameter training. From this point of view, the proposed DRME can handle more general representation learning tasks.

Table 4 lists the CPU time comparison. From the table, we can see that the proposed DRME is quite efficient when compared with KMCE and the two deep learning approaches. This phenomena demonstrates one significant merit of discarding the EM training of the base clustering in DRME.

7 Conclusions

In this paper, we have viewed several representative unsupervised representation learning algorithms as special cases of clustering ensemble. Based on this novel view, we have proposed a new deep clustering ensemble algorithm, named deep random model ensemble. In order to find the most powerful representation among the layers, we have further applied DRME to clustering. Specifically, (i) we have viewed the deep belief networks as a stack of probabilistic clustering ensemble, where each base clustering of the clustering ensemble is a binary-class probabilistic clustering. We have viewed the stacked denoising autoencoder as a stack of deterministic clustering ensemble, where each base clustering is a binary-class deterministic clustering. Moreover, when sparse coding is used as the feature encoder, we have viewed this usage of sparse coding as a kind of consensus function of clustering ensemble, and viewed the dictionary learning as the training process of the base clusterings of clustering ensemble.

TABLE 4
Comparison of the detected number of clusters. The digit in brackets is the standard deviation. The digit in italic and red color means that the corresponding method is over-fitting to the data set, hence it is meaningless. The digit in bold means that the detected number of clusters is the closest one to the true number of clusters on the data set. The column of "True number" lists true numbers of clusters. The column of "iDRME" lists the numbers of the detected clusters of the DRME that achieves the highest NMIs in Table 3. These two columns will not join in the comparison.

ID	Data	True number	KMCE	RME	PCA	DBN	SDAE	DRME	iDRME
1	Dermathology	6	5.00 (0.00)	2.00 (0.00)	2.00 (0.00)	4.95 (1.39)	2.70 (0.57)	**5.30** (1.72)	5.00 (1.97)
2	Iris	3	**3.00** (0.00)	2.00 (0.00)	2.00 (0.00)	7.25 (5.65)	2.00 (0.00)	6.45 (1.43)	2.00 (0.00)
3	Ecoli	3	**2.60** (0.49)	4.90 (0.30)	2.00 (0.00)	8.70 (4.65)	3.15 (0.37)	7.95 (3.12)	3.85 (0.59)
4	Wine	8	3.65 (0.73)	*52.85* (71.46)	2.00 (0.00)	13.65 (23.45)	15.85 (38.06)	**6.10** (1.07)	9.65 (3.44)
5	Glass	6	2.80 (1.25)	*68.70* (94.61)	3.00 (0.00)	6.25 (7.31)	3.50 (1.40)	**8.00** (1.08)	*72.75* (94.36)
6	New-Thyroid	3	**3.00** (0.00)	2.60 (0.97)	2.00 (0.00)	5.10 (3.95)	3.85 (3.36)	8.90 (2.05)	3.75 (1.94)
7	Vowel	11	2.00 (0.00)	2.95 (1.32)	6.00 (0.00)	*49.70* (121.38)	**7.85** (2.32)	15.20 (8.40)	23.60 (7.59)
8	Balance	3	*81.00* (0.00)	*312.75* (310.05)	**2.00** (0.00)	5.00 (12.25)	*623.60* (0.68)	6.95 (4.56)	*348.70* (281.64)
9	Yeast	10	3.00 (0.00)	*727.65* (725.25)	2.05 (0.22)	5.80 (8.21)	4.95 (1.57)	**10.45** (6.68)	*732.40* (739.23)
10	Satimage*	6	**4.10** (2.55)	*102.95* (434.76)	3.25 (1.67)	16.80 (40.39)	7.95 (3.89)	9.35 (5.25)	13.05 (3.12)
11	Letter*	26	2.30 (0.90)	*298.20* (703.84)	*990.40* (988.00)	*402.25* (808.63)	*1681.65* (723.79)	**4.35** (3.63)	*316.30* (714.33)
12	Pendigits*	10	2.85 (1.11)	4.30 (3.48)	3.30 (1.38)	18.85 (39.96)	4.70 (3.20)	**10.30** (6.43)	11.90 (2.43)
13	Segmentation*	7	3.05 (0.22)	3.05 (0.22)	2.85 (1.49)	2.05 (0.22)	**7.85** (4.07)	11.05 (4.67)	12.70 (2.75)
14	Optdigits*	10	3.30 (3.98)	3.60 (1.59)	2.75 (0.99)	8.00 (18.07)	*1100.90* (1018.06)	**9.70** (6.28)	11.50 (0.83)
15	Shuttle*	5	2.85 (0.85)	3.15 (1.35)	3.60 (1.28)	26.95 (75.01)	**4.65** (2.41)	13.70 (5.96)	4.50 (3.22)
16	Vehicle	4	**2.00** (0.00)	*171.05* (336.85)	**2.00** (0.00)	11.95 (22.94)	48.15 (187.57)	8.35 (3.69)	*186.20* (337.87)
17	Fea*	5	2.60 (0.86)	3.05 (2.42)	11.80 (8.52)	64.40 (186.86)	*612.45* (14.32)	**6.15** (6.38)	22.45 (5.19)
18	Libras	7	2.00 (0.00)	2.00 (0.00)	2.00 (0.00)	4.40 (2.95)	57.55 (76.38)	**6.90** (0.97)	12.55 (3.05)
19	Synthetic-Control	6	**5.00** (0.00)	2.00 (0.00)	2.00 (0.00)	2.35 (0.49)	2.00 (0.00)	11.25 (6.19)	5.10 (0.79)

(ii) Inspired by the above novel views, we might use any valid clustering ensemble to build a deep model, where the word valid means that the base clusterings should be better than the random guess and be diverse with each other. Inspired by the success of the biased approximation of the contrastive divergence learning to maximum likelihood learning for the deep belief networks, we have proposed DRME. It is a reduction of the stacked k-means ensemble to the stacked random model ensemble. A special point of the random model ensemble is that the k centers of its base clustering is k randomly sampled observations from the input observations, but not completely random ones, which accounts for the meaningfulness of the base clustering. (iii) To prevent the under-fitting and over-fitting of the learned representation to the data simultaneously, we have proposed the DRME based single-linkage clustering, where the most powerful representation is selected as the first layer that the hierarchical-tree of the single-linkage becomes stable. (iv) As a by-product, the DRME based clustering also contributes to one basic problem of clustering – detecting the natural clusters. We have conducted an extensive experiment. The experimental results have shown that the proposed DRME is more powerful than 5 state-of-the-art representation learning algorithms in terms of clustering accuracy, and moreover, it is even more powerful than the k-means clustering provided with the true number of clusters.

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *arXiv preprint arXiv:1206.5538*, 2012.

[2] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[3] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[4] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing [exploratory dsp]," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2011.

[5] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 11, no. 3, pp. 229–241, 2012.

[6] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 1096–1103.

[7] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[8] A. Coates and A. Y. Ng, "Learning feature representations with k-means," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 561–580.

[9] B. Kulis and M. I. Jordan, "Revisiting k-means: New algorithms via bayesian nonparametrics," in *Proceedings of the 29th International Conference on Machine Learning*, 2012, pp. 1–8.

[10] K. Jiang, B. Kulis, and M. I. Jordan, "Small-variance asymptotics for exponential family dirichlet process mixture models," in *Advances in Neural Information Processing Systems*, vol. 25, 2012, pp. 3167–3175.

[11] T. Dietterich, "Ensemble methods in machine learning," *Multiple Classifier Systems*, pp. 1–15, 2000.

[12] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.

[13] A. L. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, 2005.

[14] S. Vega-Pons and J. Ruiz-Shulcloper, "A survey of clustering

TABLE 5
CPU time (in seconds) Comparison. Note that the CPU time of DRME is the total training time of all 15 layers. The CPU time of the single linkage is not recorded in all methods.

ID	Data	KM*	KMCE	RME	PCA	DBN	SDAE	DRME
1	Dermathology	0.10	541.99	0.71	0.69	683.61	539.33	18.76
2	Iris	0.00	8.54	0.43	0.05	200.23	215.23	6.60
3	Ecoli	0.02	43.26	0.64	0.61	542.28	480.94	14.42
4	Wine	0.00	26.70	0.47	0.09	250.46	253.36	7.22
5	Glass	0.01	28.59	0.54	0.14	322.67	299.22	8.38
6	New-Thyroid	0.00	20.54	0.54	0.14	339.47	301.56	8.39
7	Vowel	0.24	846.02	3.21	18.43	2076.91	1413.00	113.41
8	Balance	0.01	59.46	1.92	4.12	1143.64	898.47	55.91
9	Yeast	0.29	1638.65	5.49	22.31	3336.74	3142.70	226.11
10	Satimage*	1.71	10822.57	8.05	26.16	4056.42	3040.79	387.76
11	Letter*	3.66	4936.60	8.12	26.58	4760.31	2927.43	389.86
12	Pendigits*	0.72	3608.39	7.53	26.13	4238.11	2962.09	383.33
13	Segmentation*	0.54	3828.30	7.50	26.34	3721.82	2925.62	383.26
14	Optdigits*	4.56	15873.97	7.76	26.40	5879.23	3148.21	382.39
15	Shuttle*	0.18	2214.63	8.24	26.00	4060.40	2844.46	391.10
16	Vehicle	0.11	1407.41	2.93	11.21	1622.77	1215.07	91.20
17	Fea*	1.11	8534.10	9.19	13.22	5626.56	2969.11	391.39
18	Libras	0.10	484.08	0.47	0.14	273.19	250.62	7.09
19	Synthetic-Control	0.29	2069.87	1.90	3.72	1268.45	903.38	53.54

ensemble algorithms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 03, pp. 337–372, 2011.

[15] D. Yan, A. Chen, and M. I. Jordan, "Cluster forests," *Computational Statistics and Data Analysis*, vol. PP, pp. 1–28, 2013.

[16] A. Coates and A. Y. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *Proceedings of the 28th International Conference on Machine Learning*, vol. 8, 2011, pp. 10–17.

[17] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[18] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2005, pp. 17–25.

[19] Z. He, A. Cichocki, S. Xie, and K. Choi, "Detecting the number of clusters in n-way probabilistic clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2006–2021, 2010.

[20] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystrom method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, 2004.

[21] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems*, vol. 19. MIT; 1998, 2007, p. 801.

[22] J. Han and M. Kamber, *Data mining: concepts and techniques (3rd edition)*. Morgan Kaufmann, 2011.

[23] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[24] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[25] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, 2006.

[26] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *Journal of Machine Learning Research*, vol. 2, pp. 125–137, 2002.

[27] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[28] K. Cherkauer, "Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks," in *Working Notes of the AAAI Workshop on Integrating Multiple Learned Models*, 1996, pp. 15–21.

[29] R. Maclin and J. Shavlik, "Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks," in *Proceedings of the International Joint Conference on Artificial Intelligence*, vol. 14, 1995, pp. 524–531.

[30] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Aritificial Intelligence Research*, vol. 2, pp. 263–286, 1995.

[31] T. T. Wu and K. Lange, "Coordinate descent algorithms for lasso penalized regression," *The Annals of Applied Statistics*, pp. 224–244, 2008.

[32] C. Bishop *et al.*, *Pattern recognition and machine learning*. Springer New York:, 2006.

[33] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[34] B. Schölkopf, A. Smola, and K. R. Müller, "Kernel principal component analysis," *Artificial Neural Networks*, pp. 583–588, 1997.